Subject: [PATCH -mm 1/5] swapcgroup (v3): add cgroup files
Posted by Daisuke Nishimura on Fri, 04 Jul 2008 06:17:47 GMT
View Forum Message <> Reply to Message

Even if limiting memory usage by memory cgroup,
swap space is shared, so resource isolation is not enough.
If one group uses up most of the swap space, it can affect
other groups anyway.

The purpose of swapcgroup is limiting swap usage per group
as memory cgroup limits the RSS memory usage.
It's now implemented as a add-on to memory cgroup.

This patch adds cgroup files(and a member to struct mem_cgroup)
for swapcgroup.

Files to be added by this patch are:

- memory.swap_usage_in_bytes
- memory.swap_max_usage_in_bytes
- memory.swap_limit_in_bytes
- memory.swap_failcnt

The meaning of those files are the same as memory cgroup.

Change log
v2->v3
- Rebased on 2.6.26-rc5-mm3.
v1->v2
- Rebased on 2.6.26-rc2-mm1.
- Implemented as a add-on to memory cgroup.

Signed-off-by: Daisuke Nishimura <nishimura@mxp.nes.nec.co.jp>

---
 init/Kconfig | 7 +++++
 mm/memcontrol.c | 67 ++
 2 files changed, 74 insertions(+), 0 deletions(-)

diff --git a/init/Kconfig b/init/Kconfig
index 847931a..c604b6d 100644
--- a/init/Kconfig
+++ b/init/Kconfig
@@ -418,6 +418,13 @@ config CGROUP_MEMRLIMIT_CTLR
   memory RSS and Page Cache control. Virtual address space control

```
   is provided by this controller.

+config CGROUP_SWAP_RES_CTLR
+ bool "Swap Resource Controller for Control Groups"
+ depends on CGROUP_MEM_RES_CTLR && SWAP
+ help
+   Provides a swap resource controller that manages and limits swap usage.
+   Implemented as a add-on to Memory Resource Controller.
+
 config SYSFS_DEPRECATED
  bool

diff --git a/mm/memcontrol.c b/mm/memcontrol.c
index b8fe33c..ddc842b 100644
--- a/mm/memcontrol.c
+++ b/mm/memcontrol.c
@@ -133,6 +133,12 @@ struct mem_cgroup {
   * statistics.
   */
  struct mem_cgroup_stat stat;
+#ifdef CONFIG_CGROUP_SWAP_RES_CTLR
+ /*
+  * the counter to account for swap usage
+  */
+ struct res_counter swap_res;
+#endif
 };
 static struct mem_cgroup init_mem_cgroup;

@@ -953,6 +959,39 @@ static int mem_control_stat_show(struct cgroup *cont, struct cftype *cft,
  return 0;
 }

+#ifdef CONFIG_CGROUP_SWAP_RES_CTLR
+static u64 swap_cgroup_read(struct cgroup *cont, struct cftype *cft)
+{
+ return res_counter_read_u64(&mem_cgroup_from_cont(cont)->swap_res,
+        cft->private);
+}
+
+static ssize_t swap_cgroup_write(struct cgroup *cont, struct cftype *cft,
+    struct file *file, const char __user *userbuf,
+    size_t nbytes, loff_t *ppos)
+{
+ return res_counter_write(&mem_cgroup_from_cont(cont)->swap_res,
+    cft->private, userbuf, nbytes, ppos,
+    mem_cgroup_write_strategy);
+}
```

```
+
+static int swap_cgroup_reset(struct cgroup *cont, unsigned int event)
+{
+ struct mem_cgroup *mem;
+
+ mem = mem_cgroup_from_cont(cont);
+ switch (event) {
+ case RES_MAX_USAGE:
+  res_counter_reset_max(&mem->swap_res);
+  break;
+ case RES_FAILCNT:
+  res_counter_reset_failcnt(&mem->swap_res);
+  break;
+ }
+ return 0;
+}
+#endif
+
 static struct cftype mem_cgroup_files[] = {
  {
   .name = "usage_in_bytes",
@@ -985,6 +1024,31 @@ static struct cftype mem_cgroup_files[] = {
   .name = "stat",
   .read_map = mem_control_stat_show,
  },
+#ifdef CONFIG_CGROUP_SWAP_RES_CTLR
+ {
+  .name = "swap_usage_in_bytes",
+  .private = RES_USAGE,
+  .read_u64 = swap_cgroup_read,
+ },
+ {
+  .name = "swap_max_usage_in_bytes",
+  .private = RES_MAX_USAGE,
+  .trigger = swap_cgroup_reset,
+  .read_u64 = swap_cgroup_read,
+ },
+ {
+  .name = "swap_limit_in_bytes",
+  .private = RES_LIMIT,
+  .write = swap_cgroup_write,
+  .read_u64 = swap_cgroup_read,
+ },
+ {
+  .name = "swap_failcnt",
+  .private = RES_FAILCNT,
+  .trigger = swap_cgroup_reset,
+  .read_u64 = swap_cgroup_read,
```

```
+ },
+#endif
 };

 static int alloc_mem_cgroup_per_zone_info(struct mem_cgroup *mem, int node)
@@ -1063,6 +1127,9 @@ mem_cgroup_create(struct cgroup_subsys *ss, struct cgroup *cont)
  }

  res_counter_init(&mem->res);
+#ifdef CONFIG_CGROUP_SWAP_RES_CTLR
+ res_counter_init(&mem->swap_res);
+#endif

  for_each_node_state(node, N_POSSIBLE)
   if (alloc_mem_cgroup_per_zone_info(mem, node))
```

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers